CODE 1:

def reverse\_string(s):

    reverse\_str= "" #reversed is a built\_in function to avoid potential confusion it better to use another variable name

    for i in range(len(s) - 1, -1, -1):

        reverse\_str+= s[i]

    return reverse\_str

def main():

    input\_string = "Hello, world!"

    reversed\_string = reverse\_string(input\_string)

    print(f"Reversed string: {reversed\_string}")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

CODE 2:

def get\_age():

    age = input("Please enter your age: ")

    if age.isnumeric() and int(age) >= 18: #here '>=' not supported between instances of 'str' and 'int' so here we have to convert age to int datatype

        return int(age)

    else:

        return None

def main():

    age = get\_age()

    if age:

        print(f"You are {age} years old and eligible.")

    else:

        print("Invalid input. You must be at least 18 years old.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

CODE 3:

def read\_and\_write\_file(filename):

    try:

       with open(filename, 'r') as file:

            content = file.read()

       with open(filename, 'w') as file:

            file.write(content.upper()) #this statement converts all lower case letters to uppercase letters in the file

       print(f"File '{filename}' processed successfully.")

    except Exception as e:

        print(f"An error occurred: {str(e)}")

def main():

    filename = "sample.txt" #here before executing the code we must must create a file name with sample.txt otherwise its shows error like No such file or directory: 'sample.txt'

    read\_and\_write\_file(filename)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

CODE 4:

def merge\_sort(arr):

if len(arr) <= 1:

return arr

mid = len(arr) // 2

left = arr[:mid]

right = arr[mid:]

left = merge\_sort(left) # Sort the left sub-array

right = merge\_sort(right) # Sort the right sub-array

i = j = k = 0

while i < len(left) and j < len(right):

if left[i] < right[j]:

arr[k] = left[i]

i += 1

else:

arr[k] = right[j]

j += 1

k += 1

while i < len(left):

arr[k] = left[i]

i += 1

k += 1

while j < len(right):

arr[k] = right[j]

j += 1

k += 1

return arr # Return the sorted array

arr = [38, 27, 43, 3, 9, 82, 10]

sorted\_arr = merge\_sort(arr.copy()) # Create a copy to preserve the original array

print(f"The sorted array is: {sorted\_arr}")